الدرس الأول في البايثون - Python

يحتوي الدرس الأول على مايلي:

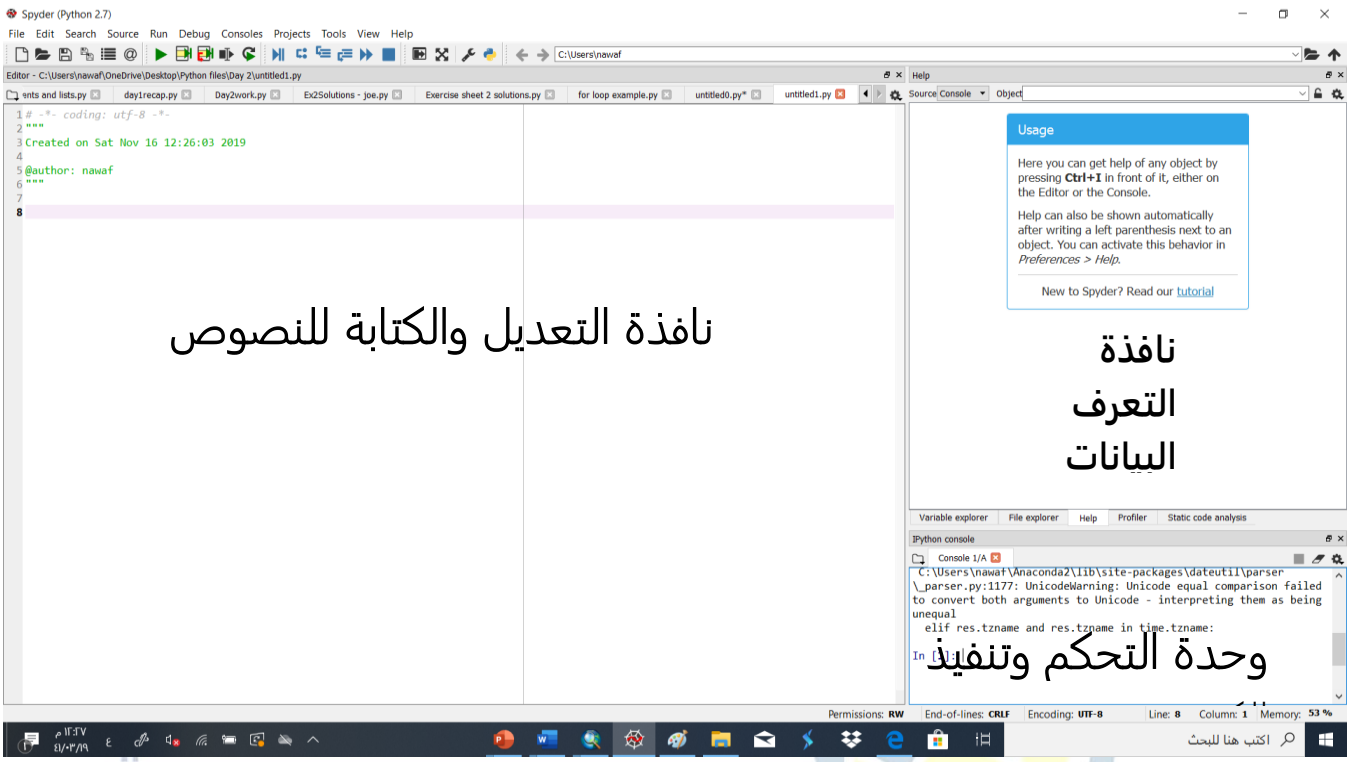
- ◀ **كيفية تنزيل البايثون Python**
- ◀ **مقدمة الى أنواع البيانات**
- ◀ **اجراء العمليات الحسابية البسيطة**
- ◀ **مقدمة الى استخدام الأداة الشرطية If**

أولا: كيفية تنزيل البايثون Python

أولا من خلال موقع www.python.org يمكن تنزيل البرنامج ولكن من افضل موزع لتنزيل البايثون ما يسمى anaconda على هذا الرابط https://www.anaconda.com/download وليكن النسخة المعروفة 2.7

سوف نستخدم بيئة تطوير مدمجة مع anaconda والذي يُعرف بـ IDE حيث ان هذا البرنامج يسمح بكتابة وتنفيذ النصوص (scripts) والذي بدوره يسهل كتابة وتنفيذا الكود في برنامج واحد. هنا سيكون الIDE هو Spyder والذي يأتي ملحق مع تنزيل Anaconda

الشكل يوضح نوافذ البرنامج Spyder



مدونة د. نواف ابراهيم العتيبي في نظم المعلومات الجغرافية و البحث العلمي

- نقطة مهمة لتفعيل النص اضغط بعده fn ومن ثم F9

المتغيرات /Variables

قيمة المتغير تُعين باستخدام العلامة "=" على سبيل المثال

A = “Nawaf”

حيث A هي المتغير و Nawaf هي القيمة التي يحتويها المتغير A

لذا يمكن كتابة الكود التالي/

```
In [7]: A = "Nawaf"
   ...: print A
Nawaf
```

بعض النقاط الهامة:

- بينما علامة "==" تعني ان المتغيِّرين متساوين
- يستخدم علامة " = " مكرر للتفريق عن الحالة السابقة
- في المقابل =! وتعني هي ان المتغيرين غير متساوين
- بامكانك ان تسمي ماتريد كمتغير variable لكن تجنب الأسماء التي لها معنى في لغة البابثون مثل list وغيرها. دائما احرص على جعل الاسم له معنى.
- اذا اردت تسمية من كلمتين فيجب ان تضع بينهما underscores "_" ليربط بينهما او التفريق بينهما بحرف انجليزي كبير مثل :

  World_club او WorldClub

ثانيا أنواع البيانات في البايثون :

- String = نص (text)
- Integer = يعني رقم (1, 2, 3 ,...)

- Float= ارقام مع فواصل عشرية مثل 3.2343
- Boolean= التعبير المنطقي True or False
- المسائل الحسابية: البايثون يمكن أن يؤدي العمليات الرياضية مثل:
- الجمع +
- الطرح –
- القسمة /
- الضرب *

مدونة
د. نواف ابراهيم العتيبي
في نظم المعلومات الجغرافية
و البحث العلمي

ثالثا امثلة على العمليات الحسابية البسيطة :

```
In [24]: print 5**2
25

In [25]: print 6/2
3

In [26]: print 4**4
256

In [27]: print 5**2
25

In [28]: print 4**4
256

In [29]: print 4.0 **2
16.0
```

عندما تكون هناك عمليات حسابية يتم استيراد أدوات الرياضيات كالتالي:

"import math"

```
In [7]: import math
   ...:
   ...: print math.sqrt(16)
4.0
```

رابعا استخدام العبارة الشرطية if:

- تشغيل الكود اذا تحقق الشرط (if) على سبيل المثال:

```
In [10]: a = 5
    ...: b = 6
    ...: if a != b:
    ...:     print a
    ...:
    ...:
5
```

- المسافة البادئة مهمة جدا في كتابة في البايثون لذا الكود اللي بعد عبارة if يجب ان تكون هناك مسافة للدلالة على تبعيته للعبارة الشرطية كما هو في المثال السابق.

مدونة
د. نواف ابراهيم العتيبي
في نظم المعلومات الجغرافية
و البحث العلمي

```
a = 5

b = 5

if a != b:

    print a
```

نقطتين

مسافة

نستكمل ما يتعلق بالعبارة الشرطية "if" في الدرس القادم ان شاء الله

اخوكم

د. نواف العتيبي

مدونة
د. نواف ابراهيم العتيبي
في نظم المعلومات الجغرافية
و البحث العلمي

مدونة
د. نواف ابراهيم العتيبي
في نظم المعلومات الجغرافية
و البحث العلمي

الدرس الثاني في البايثون – Python

العبارات الشرطية if

- نستطيع استخدام elif بهدف التأكد من أن هناك شرط اخر قد يكون صحيح اذا لم يتحقق الشرط الأول باستخدام if
- كما يمكن استخدام else والتي تضيف عبارة بأن اذا لم يتحقق الشرط الأول الموضح في if ولا الشرط الثاني في elif فانه يتحقق الظرف الثالث في else

المثال التالي:

مدونة
د. نواف ابراهيم العتيبي
في نظم المعلومات الجغرافية
و البحث العلمي

```
In [3]: C = 7
   ...: if C == 5:  #  5 تحقق هل مساوي
   ...:     print "C is 5"
                      c # اذا تحقق الشرط سيتم طباعة      :...
   ...: elif C > 5:  #اذا لم يكون مساوي للخمسة تحقق هل هو اكبر من
خمسة
   ...:     print "Print C is more than 5"
   ...: else:  # اذا لم يتحقق ولا واحد من الظرفين اعلاه فاطبع التالي
   ...:     print "C is less than 5"
   ...:
   ...:
Print C is more than 5

In [4]:
```

تمرين باستخدام عبارات if:

- اكتب كود Code يجعل المتقدم على الوظيفة يقبل الوظيفة اذا كانت في الرياض "Riyadh" والراتب salary اعلى من ٢٠ الف ريال اما اذا كانت الوظيفة في مدينة أخرى فانه يقبلها اذا كان الراتب الشهري اكثر من ١٠ الف ريال غير ذلك فانه لا يقبل الوظيفة (اذا قبول الوظيفة يعتمد على الموقع و مقدار الراتب)؟

```
In [7]: location = raw_input('Please enter name of the city')
   ...: location = location.strip()
   ...: salary = raw_input('Please enter the salary')
   ...: salary = float(salary)
   ...:
   ...:
   ...: if location == 'Riyadh' and salary > 20000:
   ...:     print 'Accept the job in %s' %(location)
   ...: elif location != 'Riyadh' and salary > 10000:
   ...:     print 'Accept the job in %s' %(location)
   ...: else:
   ...:     print 'I do not accept the job'
   ...:
   ...:

Please enter name of the city Taif
Please enter the salary 12000
Accept the job in Taif
```

هنا تم اخال اسم المدينة

هنا تم ادخال مقدار

هنا نتيجة تنفيذ الكود

نستكمل في الدرس القادم ان شاء الله التعامل مع القوائم Lists و indexing

اخوكم

د. نواف العتيبي

الدرس الثالث في البايثون – Python

التعامل مع القوائم List, indexing

- القائمة(list) هي نوع البيانات التي تحتوي قيم متعددة. على سبيل المثال اذا اردت ان تجمع عدة نصوص او ارقام في متغير واحد مستقل
- يتم العريف بها باستخدام الاقواس [ ] والقيم التي بداخلها يتم الفصل بينها بالفواصل. على سبيل المثال:

List = [4, 6, 8] or my_list = ['Riyadh', 'Jeddah', 'Dammam']

- يمكن ان تحتوي هذه القائمة أي نوع من البيانات وكذلك يمكن ان تحتوي قوائم أخرى
- تحديث القوائم lists قد يكون ذلك باستخدام "append" والتي تعني الحاق/إضافة على سبيل المثال:

```
In [20]: list = [4,6, 8]

In [21]: list.append(10)

In [22]: print list
[4, 6, 8, 10]
```

- بالنسبة لما يسمى Indexing فهي طريقة لاختيار أجزاء من قائمة list بناء على الموقع على سبيل المثال: اختيار العنصر الثالث او الرابع في القائمة
- من المهم التوضيح بأن التسلسل في الترتيب في لغة البايثون يبدأ بالصفر 0 ومن ثم ١ وهكذا لذا اذا اردنا اختيار الرقم الأول يكون يتحديد 0
- فالاختيار العناصر في البايثون يكون بوضع الاقواس المربعة [] مثال على ذلك:

```
In [26]: my_list = ['Riyadh', 'Jeddah', 'Dammam']

In [27]: print my_list[0]
Riyadh
```

- يمكن استخدام indexing لاختيار اكثر من عنصر في القائمة كما في المثال التالي:

```
In [29]: my_list = ['Riyadh', 'Jeddah', 'Dammam']

In [30]: print my_list[:2]
['Riyadh', 'Jeddah']
```

- كما يمكن استخدام indexing بالعكس لاختيار اخر عنصر في القائمة مثلا كما في المثال التالي:

```
In [31]: my_list = ['Riyadh', 'Jeddah', 'Dammam']

In [32]: print my_list[-1]
Dammam
```

نستكمل في الدرس القادم ان شاء الله Replace و For

اخوكم

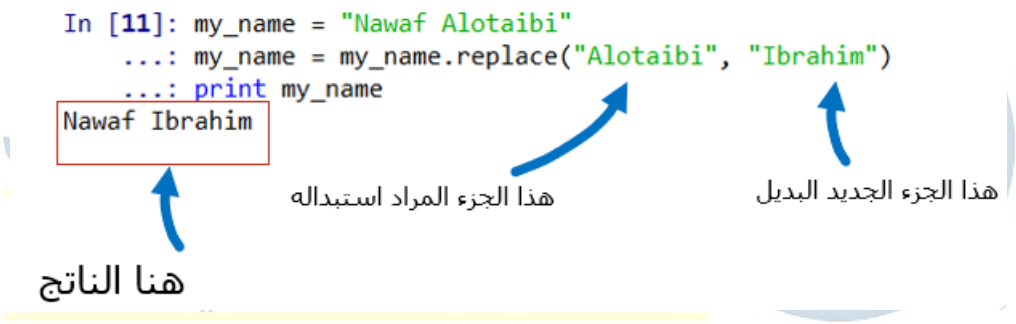
د. نواف العتيبي

الدرس الرابع في البايثون – Python

الاستبدال replace و الحلقات (التكرار) loop

أولا: الاستبدال replace

- استبدال أجزاء من النصوص باستخدام replace
- مثال:



الحلقات والتكرار For loops

- استخدام for للتكرار يسمح لتنفيذ العمليات على كل عنصر من ضمن قائمة
- مثال اذا كان لديك قائمة بأعداد وتريد ان تطبع النتيجة بعد إضافة عدد مثلا 3 لكل رقم "عدد" في المجموعة كما في المثال التالي:

```
In [15]: List1 = [3, 5, 7, 9]

In [16]: for i in List1:
    ...:     print i+3
    ...:
    ...:
6
8
10
12
```

- لتحديث العناصر في قائمة لديك يمكن ذلك بسهوله عن طريق انشاء قائمة جديدة تحتوي العناصر المحدثة كما في المثال التالي:

```
In [19]: List1 = [3, 5, 7, 9]
    ...: New_List1 = []
    ...:
    ...: for i in List1:
    ...:     New_List1.append(i+3)
    ...:
    ...: print New_List1
[6, 8, 10, 12]
```

هنا القائمة تم تحديثها

تم استخدام append للإضافة

اخوكم

د. نواف العتيبي

الدرس الخامس في البايثون – Python

While - حلقات (التكرار) loop

- While وتعني بالعربية "طالما" و loops وتعني حلقات تكرار وهذا يعني انا استخدام مفردة While في الكود مشابه لاستخدام For والتي تم شرحها سابقا لكن هناك فرق جوهري

مهم :

- هو أن استخدام While يعني ان "loop" تكرار الأوامر سوف يستمر في العمل فقط "طالما" الشرط المحدد في الكود صحيح!

مثال للتوضيح:

طالما X اصغر من 50 فاطبع قيمة x واستمر في ذلك... حتى تكون قيمة x اكبر من 50

```
In [14]: x=25
    ...:
    ...: while x < 50:
    ...:     print x
    ...:     print 'x is still less than 50, Please continue'
    ...:     x = x + 3
    ...:
    ...:
25
x is still less than 50, Please continue
28
x is still less than 50, Please continue
31
x is still less than 50, Please continue
34
x is still less than 50, Please continue
37
x is still less than 50, Please continue
40
x is still less than 50, Please continue
43
x is still less than 50, Please continue
46
x is still less than 50, Please continue
49
x is still less than 50, Please continue
```

اخوكم د. نواف العتيبي

الدرس السابع في البايثون – Python

التعريف بالدالة Function /المهمة او الوظيفة

- مع تنفيذ عمليات محددة على عدد من المتغيرات بنفس طريقة فان ذلك يكون ممكنا اذا كانت هذه العملية بسيطة ولكن قد تصبح العملية معقدة اذا كانت كثيرة ومتداخلة لذا من الأفضل استخدام ما يسمى Function لإعادة وتكرار العملية اكثر من مرة . على سبيل المثال:

✓ اذا انت تريد ان تأخذ قائمة وتقوم بضرب أي رقم زوجي فيه بعدد 2 وهذا يتطلب ان تستخدم كود ليتم تنفيذه مرة واحدة

✓ قبل تنفيذ العملية يتم ذلك بعمل قائمة بالأرقام ويمكن استخدام خاصية range (0, 10) ومن ثم يتم استخدام For

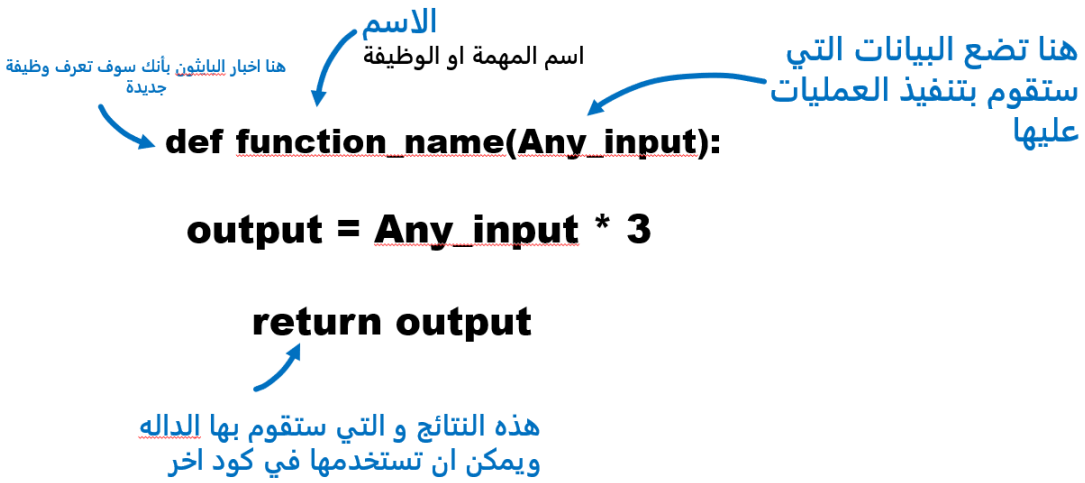
- ليتم عملية التكرار Loop واستخدام الأداة الشرطية if من اجل تنفيذ الشرط بأن اذا الرقم زوجي في القائمة فأنه يتم ضربه في عدد 2:

```
In [10]: my_list = [1,4,5,8,9,7]

In [11]: for index, item in enumerate(my_list):
    ...:     if item%2 == 0:
    ...:         my_list[index] *=2
    ...:
    ...: print my_list
[1, 8, 5, 16, 9, 7]
```

- بكل تأكيد اذا كنا نريد العديد من القوائم Lists والتي نريد ان نقوم بتنفيذ نفس العملية عليها لذا سيكون من الصعب إعادة نفس العملية في كل مره تكتب نفس الكود!

- لذا من الأفضل التعريف بالدالة Function والتي يمكن ان نعيد استخدامها! كما في الشكل المبين بالأسفل.



في الكود التالي تم عمل دالة (مهمة/function) مهمتها التأكد من أن الاعداد الزوجية في القائمة مضروبة مرتين وإذا كان غير ذلك (فردي) فانه تجعل العدد كما هو في القائمة.

د. نواف ابراهيم العتيبي

```
In [21]: list1 = [1,3,5,7,8,10]

In [22]: def double(list1):
    ...:     for index, item in enumerate(list1):
    ...:         if item%2 == 0:
    ...:             list1[index] *=2
    ...:     return list1
    ...:
    ...:
    ...: double (list1)
Out[22]: [1, 3, 5, 7, 16, 20]
```

في هذا يتم عمل التكرار على القائمة list1 وضرب كل عدد زوجي في 2 ومن ثم إضافته الى القائمة وهكذا يتم تحديث القائمة

اخوكم

د. نواف العتيبي

الدرس الثامن في البايثون - Python

قراءة الملفات Reading files

- بإمكانك فتح الملف باستخدام البايثون والتعامل معه كمكون والذي يمكن ان يتم معالجته بشكل مستقل عن الملف الأصلي
- كائن الملف يمكن ان يُتعامل معه باستخدام الدالة open ()
- دالة فتح الملف تفترض ان اسم الملف هو أول شيء يُتعامل معه في الكود
- المثال الذي بالاسفل يوضح كيفية انشاء ملف باستخدام البايثون بمسمى my_file والذي يمثل محتوى للملف "houses.txt" كما في مثال 1
- الملف يتطلب ان يُحفظ في نفس الموقع الذي يُحفظ في ملف البايثون py. والذي يتم العمل عليه

بدون تكملة لهذا الكود , البايثون سيفترض انك تريد قراءة الملف "r" كما في مثال 1

مثال 1

```
my_file = open('houses.txt', 'r')
```

- هناك ثلاث طرق لقراءة البيانات في مكون الملف- read (), readline() and readlines()
- أولا فيما يتعلق بـ()read تدل على قراءة جميع محتوى الملف كنص مفرد – واحد كما في المثال التالي:

```
my_file = open('houses.txt', 'r')
housestxt = my_file.read()
```

- هنا بعض الأمثلة من الأوامر والتي يمكن تنفذ على كامل البيانات مرة واحدة لقراءتها كما في المثال التالي:

مثال 2:

```
12 my_file = open('houses.txt', 'r') #فتح الملف, وقراءته
13 housestxt = my_file.readlines() #قراءة المحتوى والبدء في قراءة القائمة من سطر
14 print type(housestxt) #طباعة نوع المتغير
15 print len(housestxt) #طباعة عدد العناصر في القائمة بالملف
16 print housestxt[4] #طباعة قيمة العنصر الرابع في مجموعة القائمة
17 my_file.close()
18
```

من الأفضل دائما اقفال الملف بعد الانتهاء من العمل عليه وذلك باستخدام "اسم الملف". Close . بعض الأحيان التغييرات التي يتم اجراؤها على الملف لاتتم الا بعد اقفاله!

النتائج:

```
<type 'list'>
41
Houses were built in best locations.
```

تمرين:

- في هذا التمرين سيتم فتح ملف houses.txt وتعيينه كأسم متغير من اختياري
- استخدام read () لقراءة محتوى الملف كنص
- استخدام forحلقات التكرار والعمل على معرفة كم تكرر حرف "b" في النص - الملف

هنا تحتاج الى انشاء متغير يبدأ من صفر 0 ومن ثم loop على كل رمز في النص ومن ثم تغيير هذا الرقم 0 كلما مر على حرف b

```
In [6]: my_file = open('houses.txt', 'r')
   ...: housestxt = my_file.read()
   ...: my_file.close()
   ...:
   ...: count = 0
   ...: for i in housestxt:
   ...:     if i.lower() == 'b':
   ...:         count = count + 1
   ...:
   ...: print count
13
```

– ناتج عمل الكود هو 13 مرة تكرر حرف b

اخوكم

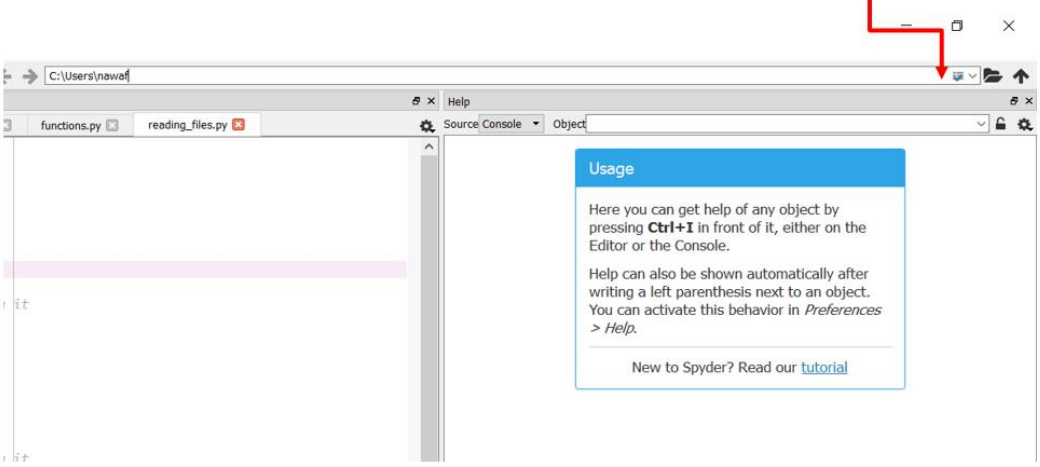
د. نواف العتيبي

الدرس التاسع في البايثون – Python

التعامل مع وحدة CSV – الجزء الأول (مقدمة مختصرة الى مايسمى CSV module)

- سوف نلقي الضوء على قراءتها و كيفية التعامل مع الصفوف في هذا النوع من الملفات وكذلك كيفية إنشاء بيانات جديدة من ملفات CSVs موجودة.
- من الأفضل دائما حفظ ملف py (امتداد ملفات البايثون) في نفس المجلد المحفوظ فيه ملف CSV المستخدم.
- كذلك من الأفضل ان يكون مكان امتداد حفظ الملف هو المكان الذي تعمل عليه!
- مع Spyder (بيئة تطوير مدمجة مع anaconda والذي يُعرف بـ IDE) يمكن بسهولة تحديد الموقع الذي يتعامل معه البايثون بشكل مباشر و بسهوله من خلال هذه الايقونة الموضح في الشكل بالاسفل:

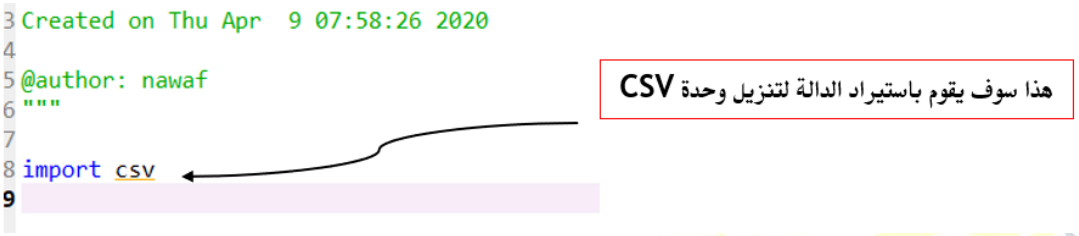
امتداد الملف واعداده يتم هنا



أولا استيراد وحدة CSV

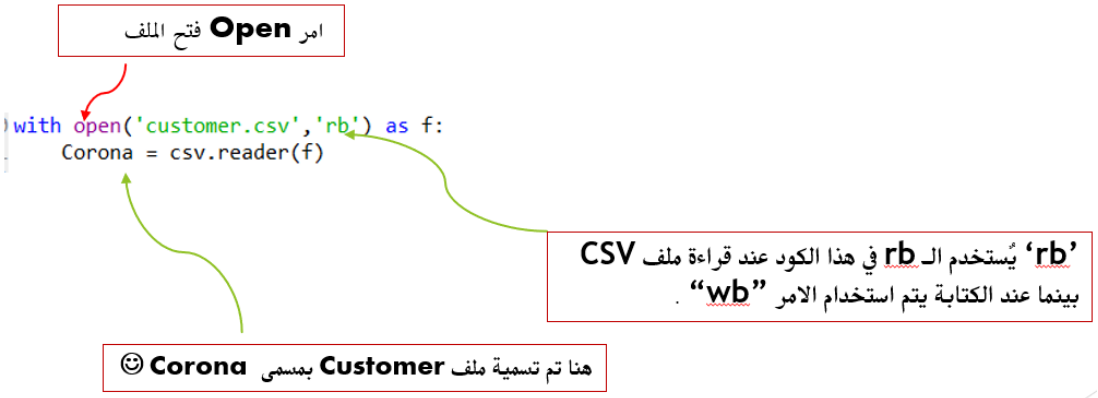
- **هذه الوحدة تقدم خيارات متعددة من تنزيل ومعالجة وكتابة ملفات CSV**

ويتم تنزيلها كالتالي:



ثانيا : انشاء دالة كائن القراءة Reader

- **هذا الشكل من الكود يفتح ملف CSV ويُنشا مايسمى بالقارئ Reader**



ثالثا: التكرار loop خلال CSV

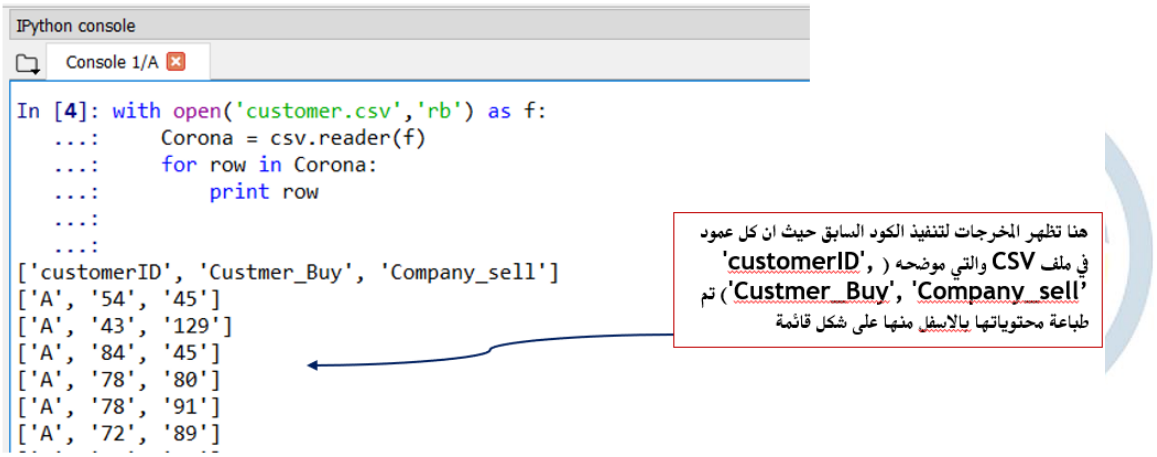
- **استخدام الدالة Reader للقراءة كل سطر في الملف الذي تم إنشاءه باسم Corona –(😊) وطباعته عن طريق استخدام الدالة For loop**

استخدام الدالة For للتكرار عند قراءة كل سطر في الملف

```
10 with open('customer.csv','rb') as f:
11     Corona = csv.reader(f)
12     for row in Corona:
13         print row
```

هذا سوف يطبع كل صف في الملف كما هو

المخرجات :



سوف سنتكمل الدرس الأخير مع مثال بإذن الله

اخوكم د. نواف العتيبي

الدرس العاشر في البايثون – Python

التعامل مع وحدة CSV – الجزء الثاني

تنفيذ الأوامر وحلقات التكرار من خلال ملف CSV

- نستطيع تجاوز الصف الأول First row في ملف CSV على سبيل المثال أسماء العنوانين في الاعمدة باستخدام

Next (f) كما في المثال التالي:

```
 8 import csv
 9
10 with open('customer.csv','rb') as f:
11     Corona = csv.reader(f)
12     next (f)
13     for row in Corona:
14         print row
```

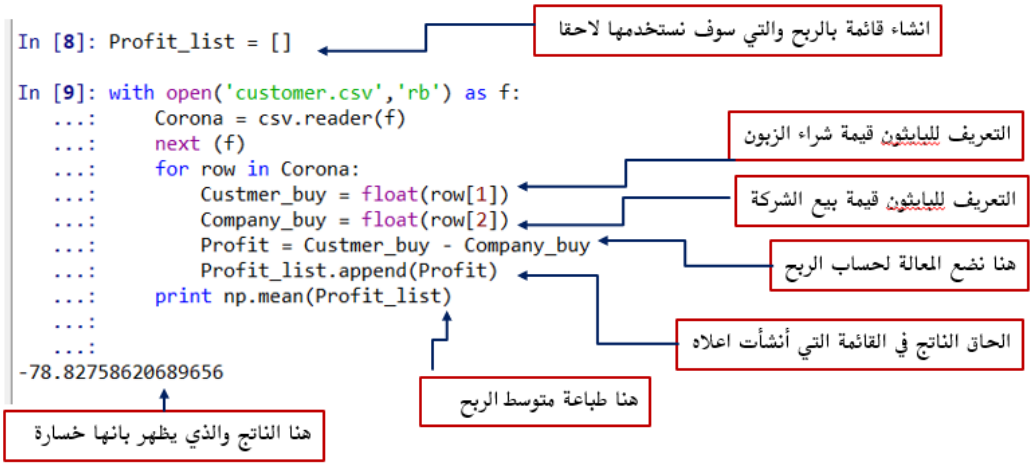
هنا تنفيذ الكود:

```
...:
['A', '54', '45']
['A', '43', '129']
['A', '84', '45']
['A', '78', '80']
['A', '78', '91']
['A', '72', '89']
['A', '72', '77']
['A', '66', '456']
```

- من المكتبات الهامة التي سوف نقوم باستيرادها هي numpy لتنفيذ الدوال الإحصائية كما في الشكل التالي:

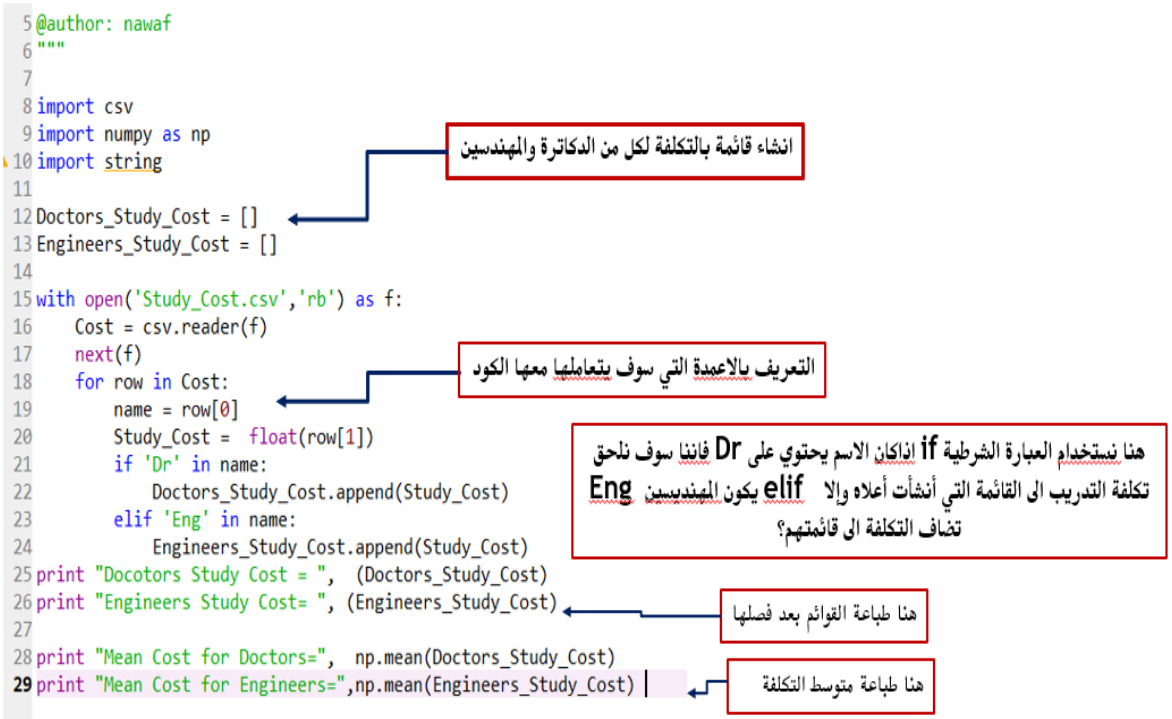
```
10 import numpy as np
```

- مثال ١ : سنحاول في الملف السابق Customer.csv ان نحسب الفرق بين قيمة شراء السلعة من البائع وقيمتها عند البيع على المستهلك من اجل حساب الربح ومن ثم حساب متوسط الربح بشكل عام.



- **مثال ٢: في التمرين التالي سوف نستخدم العبارة الشرطية if و elif**

✓ **لدينا ملف Study_Cost.csv يحتوي على قائمة بالأسماء ملحق في كل اسم لقب Dr او Eng (مثلا Dr. Mohammed او Eng. Saud) وتكلفة التدريب السنوية لكلا منهم:**



ونريد من البايثون قراءة الصفوف loop في هذه القائمة وفصل بين الدكاترة والمهندسين ومن ثم انشاء قائمة لكل فئة ومن ثم حساب متوسط التكلفة لكل منهم؟

```
In [46]: with open('Study_Cost.csv','rb') as f:
    ...:     Cost = csv.reader(f)
    ...:     next(f)
    ...:     for row in Cost:
    ...:         name = row[0]
    ...:         Study_Cost =  float(row[1])
    ...:         if 'Dr' in name:
    ...:             Doctors_Study_Cost.append(Study_Cost)
    ...:         elif 'Eng' in name:
    ...:             Engineers_Study_Cost.append(Study_Cost)
    ...:
    ...: print "Docotors Study Cost = ",  (Doctors_Study_Cost)
    ...: print "Engineers Study Cost= ", (Engineers_Study_Cost)
    ...:
    ...: print "Mean Cost for Doctors=",  np.mean(Doctors_Study_Cost)
    ...: print "Mean Cost for Engineers=",np.mean(Engineers_Study_Cost)
Docotors Study Cost =  [20000.0, 30000.0, 30000.0, 20000.0]
Engineers Study Cost=  [50000.0, 60000.0, 40000.0]
Mean Cost for Doctors= 25000.0
Mean Cost for Engineers= 50000.0
```

هنا نتائج تنفيذ الكود

والحمدالله على ختام المقدمة الى البايثون,,,

اخوكم د. نواف العتيبي